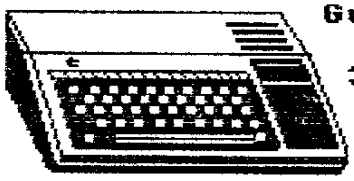
GUILFORD 99'ERS NEWSLETTER

SUPPORTING THE TEXAS INSTRUMENTS TI-99/4A COMPUTER



GUILFORD 99'ERS UG
3202 CANTERBURY DR
GREENSBORO NC
27408

TO:

VOLUME 9 NUMBER 1 JANUARY - MARCH 1992

, Pres. Bob Carmany, Newsletter Ed (855-1538)
Tony Kleen, Sec/Treas (924-6344) Bill Woodruff, Pgm/Library (228-1892)
BBS: (919)621-2623 --ROS

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to:Tony Kleen c/o 3202 Canterbury Dr., Greensboro, NC 27408. The Software Library is for dues paying members only. (Bob Carmany Ed)

OUR NEXT MEETING

DATE: December 3,1991 Time: 7:30 PM. Place: Glenwood Recreation Center, 2010 S. Chapman Street.

Program for this meeting will be our Christmas program --sounds of the season, good cheer, and the "odd" surprise thrown in for good measure. Be sure to be there and bring a "munchie" with you and we'll share the lot with everyone!

MINUTES

The November meeting of the Guilford 99'er Users's Group was held on Tuesday the 12th at the Glenwood Recreation Center. There were five members present.

The Treasurer reported that we have $137.49 in the treasury. We donated $25 to the American Heart Association in memory of our late President, George von Seth.

There were not minutes from last month's meeting since there was no business transacted at that meeting. We just went right into the program (re: The GRAMKRACKER). Thanks to Roy West and Mac for their insight. Thanks for Bob for his overview of Funnelweb.

Bob brought it to our attention that The Hunter Valley UG has been resurrected. Contact Geoff Phillips, New Hunter Valley Computer Club, PO Box 347, Warners Bay, NSW 2282, Australia for details. Other news is that the periennial final (?) version of Funnelweb 4.40 is out with yet another version promised.

The program for the evening was presented by Tony Kleen. He demonstarted how to use TI-Base to keep track of installment loans. Once again, an excellent program.

Respectfully submitted,

Tony Kleen Sec/Treas

EDITOR'S NOTES

The December issue marks the end of the monthly newsletters for the Users' Group. Input for the newsletter has dwindled to near nothing and it is virtually impossible to continue at the current schedule. Beginning in January, the newsletter will be published quarterly and it is anticipated that the length will be increased as well. Maybe with a couple of months between issues, we can get enough to fill 18 or so pages. Let's hope!

Remember, your annual dues are due and collectable in January as well. Bring your $$$ to the December or mail your check to the UG mailing address on the newsletter. Come back for another year --- we ain't finished yet!

In fact, read this next bit if you think your TI is primitive when compare to the (ugh!) IBM.

UP YOURS---IBM

By Bob Carmany

I ran across this little "gem" while going through our newsletter at work. MultiMate is the word processor that is used frequently in the IBM PS/2 world. Just read the instructions as they are written for the IBM and compare them with those in the next section for F'WEB and the TI.

"Say, for example, that you are creating a document and would like it to contain text that already exists in another, previously created document. Instead of typing the text into the new document, you can simply copy it from the existing document. Here is how. . .

Using the "Create a Document" or "Edit a Document" commands from the MultiMate Main Menu, place on the screen the document into which you wish to copy the text (I will call this the target document).

Next, use the arrow keys to move the cursor in the target document to the location where you want the copied text to appear.

Hold down the SHIFT key and tap the F8 key. The target document will temporarily disappear from the screen and a document directory will appear. The words "EXTERNAL COPY" should appear in the upper right-hand corner of the screen.

Use the F4 and F5 keys to change drives and paths as needed to locate the document from which you wish to copy the text (I will call this document the source document).

Use the arrow keys to move the red highlight to the source document file name and tap the F10 key. The first page of the source document should appear on the screen. The words "START COPY WHERE" should appear in the upper right-hand corner of the screen.

Use the arrow keys to move the cursor to the beginning of the text that you want to copy and tap the F10 key. The words "COPY WHAT?" should appear in the upper right-hand corner of the screen.

Use the arrow keys to move the cursor to the end of the text that you want to copy and tap the F10 key. The source document should then disappear from your screen and the target document should reappear with the copied text in the correct location."

Now, for the TI version of the same operation using the same terminology:

Load the target document into F'WEB. Hit FCTN-9 and LF. The format for "external copy" is [start line] [end line] [after line] DSKx.filename. The first two entries can be omitted if you wish to copy the entire source text file. You can also use "E" for the third entry if you wish for the source text to be added to the end of the target file.

It's a lot easier that all of the "garbage" we went through on the IBM. Here are a couple of examples:

0001 0100 0200 DSKx.Filename copies lines 0001 through 0100 after line 200 in the text.

0050 0075 E DSKx.filename copies lines 0050 through 0075 at the end of the target file.

E DSKx.filename copies the entire source file at the end of the target file.

UP YOURS----IBM!!!

```
100 REM ********************
110 REM * BARS AND GRAPHS *
120 REM ********************
130 REM BBY JOHN GUNTER
140 REM HOME COMPUTER MAGAZI
NE
150 REM VERSION 4.3.1
160 REM TI BASIC
170 CALL CLEAR
180 PRINT TAB(7);"BARS AND G
RAPHS": : : : : : :
190 INPUT "USE A PRINTER (Y/
N)?":P$
200 IF (P$<>"Y")&(P$<>"N")TH
EN 190
210 IF P$="N" THEN 290
220 PRNT=1
230 INPUT "PRINTER DEVICE NA
ME?":DEV$
240 GOTO 290
250 CALL KEY(0,K,S)
260 IF (S=0)+(K<>13))THEN 25
0
270 CALL CLEAR
280 RETURN
290 PRINT TAB(10);"PLEASE WA
IT": :
300 PRINT TAB(12);"GOTTA RES
T"
310 DIM CO(32)
320 DIM A$(11)
330 A$(1)="00000000000007E"
340 A$(2)="0000000000007E7E"
350 A$(3)="000000000007E7E7E"
360 A$(4)="00000007E7E7E7E"
370 A$(5)="0000007E7E7E7E7E"
380 A$(6)="00007E7E7E7E7E7E"
390 A$(7)="007E7E7E7E7E7E7E"
400 A$(8)="7E7E7E7E7E7E7E7E"
410 A$(9)="0101010101010101"
420 A$(10)="0101010101010FF
"
430 A$(11)="FF01010101010101
"
440 CALL SCREEN(13)
450 CALL COLOR(9,5,1)
460 CALL COLOR(10,9,1)
470 CALL COLOR(11,11,1)
480 CALL COLOR(12,15,1)
490 COL(1)=5
500 COL(2)=9
510 COL(3)=11
520 COL(4)=15
530 FOR X=95 TO 158 STEP 8
540 FOR CH=1 TO 8
550 CALL CHAR(X+CH,A$(CH))
560 NEXT CH
570 NEXT X
580 CALL CHAR(42,"935539FF39
5593FF")
590 CALL CHAR(91,"0000000000
0000FF")
600 CALL CHAR(92,A$(9))
610 CALL CHAR(93,A$(10))
620 CALL CHAR(94,A$(11))
630 CALL CLEAR
640 SET=0
650 CALL VCHAR(1,7,92,21)
660 FOR X=1 TO 21
670 CALL HCHAR(X,8,91,25)
680 NEXT X
690 CALL HCHAR(21,8,93,24)
700 FOR TIC=5 TO 20 STEP 5
710 CALL HCHAR(TIC,7,93)
720 NEXT TIC
730 CALL HCHAR(1,7,94)
740 ST$="MAXIMUM DATA VALUE
? "
750 GOSUB 770
760 GOTO 1190
770 D1$=D$
780 D$=""
790 FOR ST=1 TO LEN(ST$)
800 CALL HCHAR(24,2+ST,ASC(S
EG$(ST$,ST,1)))
810 NEXT ST
820 KI=1
830 CALL SOUND(100,800,2)
840 CALL HCHAR(23,7,42)
850 CALL KEY(0,K,S)
860 IF S=0 THEN 850
870 IF K=13 THEN 1140
880 IF K=7 THEN 1080
890 IF K=8 THEN 980
900 IF K=9 THEN 930
910 IF K<>32 THEN 1030
920 IF (ST$="SIDE LABEL ? ")
+(ST$="BOTTOM LABEL ? ")+(ST
$="LEGEND 1")+(ST$="LEGEND 2
")THEN 1030
930 IF M+1>32 THEN 840
940 CALL HCHAR(21,M,93)
950 M=M+1
960 CALL HCHAR(21,M,42)
970 GOTO 840
980 IF M-1<8 THEN 840
990 CALL HCHAR(21,M,93)
1000 M=M-1
1010 CALL HCHAR(21,M,42)
1020 GOTO 840
1030 IF KI>25 THEN 840
1040 CALL HCHAR(23,7+KI,K)
1050 D$=D$&CHR$(K)
1060 KI=KI+1
1070 GOTO 840
1080 CALL SOUND(300,300,2)
1090 FOR ES=KI TO 1 STEP -1
1100 CALL HCHAR(23,7+ES,32)
1110 NEXT ES
1120 D$=""
1130 GOTO 820
1140 CALL HCHAR(23,1,32,32)
1150 CALL HCHAR(24,1,32,32)
1160 IF D$<>"" THEN 1180
1170 D$=D1$
1180 RETURN
1190 FOR X=1 TO LEN(D$)
1200 IF (ASC(SEG$(D$,X,1))<4
8)+(ASC(SEG$(D$,X,1)))>57)THE
N 1240
1210 NEXT X
1220 IF (LEN(D$)<1)+(LEN(D$)
>5)THEN 1240
1230 GOTO 1260
1240 GOSUB 1870
1250 GOTO 740
1260 MD=VAL(D$)
1270 SCA=(INT(MD/20)*20)+20
1280 IF INT(MD/20)<>MD/20 TH
EN 1300
1290 SCA=SCA-20
1300 DIV=100
1310 LA$=STR$(SCA)
1320 FOR MK=1 TO 21 STEP 5
1330 FOR LAB=1 TO LEN(LA$)
1340 CALL HCHAR(MK,2+LAB,ASC
(SEG$(LA$,LAB,1)))
1350 NEXT LAB
1360 DIV=DIV-25
1370 SCA1=SCA*(DIV/100)
1380 LA$=STR$(SCA1)
1390 NEXT MK
1400 ST$="SIDE LABEL ? "
1410 GOSUB 770
1420 IF LEN(D$)<25 THEN 1440
1430 D$=SEG$(D$,1,24)
1440 FOR LAB=1 TO LEN(D$)
1450 CALL VCHAR(LAB,2,ASC(SE
G$(D$,LAB,1)))
1460 NEXT LAB
1470 ST$="BOTTOM LABEL ?"
1480 GOSUB 770
```

```
1490 R=22
1500 Y=7
1510 GOSUB 1530
1520 GOTO 1570
1530 FOR LAB=1 TO LEN(D$)
1540 CALL HCHAR(R,Y+LAB,ASC(
SEG$(D$,LAB,1)))
1550 NEXT LAB
1560 RETURN
1570 M=0
1580 ST$="COLOR?1-BLU,2-RED,
3-YEL,4-GY"
1590 GOSUB 770
1600 IF (ASC(D$)<49)+(ASC(D$
)>52)+(LEN(D$)>1)THEN 1580
1610 C=VAL(D$)-1
1620 IF SET=0 THEN 1770
1630 CH=127
1640 CH1=135
1650 SE=SET-1
1660 CST=13+SET-1
1670 IF CST<17 THEN 1790
1680 CST=CST-1
1690 SE=SE-1
1700 CALL SOUND(300,300,2)
1710 D$="COLOR FULL-ENTER ,ER
ASE OR END"
1720 Y=1
1730 R=24
1740 GOSUB 1530
1750 ST$=""
1760 GOTO 1800
1770 CH=95
1780 CH1=103
1790 ST$="ENTER DATA"
1800 CALL HCHAR(21,M,42)
1810 CALL HCHAR(21,M-1,93)
1820 GOSUB 770
1830 IF D$<>"C" THEN 1900
1840 IF SET=0 THEN 1580
1850 SET=SET+1
1860 GOTO 1580
1870 CALL SOUND(300,300,2)
1880 ST$="BAD DATA-TRY AGAIN
"
1890 RETURN
1900 IF D$="END" THEN 2330
1910 IF D$<>"ERASE" THEN 196
0
1920 CALL HCHAR(21,M,93)
1930 CALL VCHAR(2,M,91,19)
1940 CALL HCHAR(1,M,95)
1950 GOTO 1770
1960 IF (D$="P")&(PRNT=1)THE
N 2560
1970 IF D$="NEW" THEN 630
1980 FOR X=1 TO LEN(D$)
1990 D=ASC(SEG$(D$,X,1))
2000 IF (D<47)+(D>57)THEN 20
30
2010 NEXT X
2020 GOTO 2050
2030 GOSUB 1870
2040 GOTO 1820
2050 D=VAL(D$)
2060 IF D>SCA THEN 2080
2070 GOTO 2100
2080 GOSUB 1870
2090 GOTO 1820
2100 BAR=D/(SCA/20)
2110 HT=INT(BAR)
2120 RE=BAR-HT
2130 TP=1+INT((RE*8)+.5)
2140 IF TP<9 THEN 2160
2150 TP=8
2160 SE=C
2170 CALL CHAR(CH+TP+(SE*8),
A$(TP))
2180 CALL CHAR(CH1+(SE*8),A$
(8))
2190 CALL VCHAR(21-HT,M,CH1+
(SE*8),HT)
2200 CO(M)=C
2210 IF HT>19 THEN 2250
2220 CALL HCHAR(20-HT,M,CH+T
P+(SE*8))
2230 CALL VCHAR(1,M,91,19-HT
)
2240 GOTO 2260
2250 CALL HCHAR(21-HT,M,CH1)
2260 IF M<32 THEN 2310
2270 CALL SOUND(300,300,2)
2280 ST$="DATA FULL"
2290 GOSUB 770
2300 GOTO 1790
2310 M=M+1
2320 GOTO 1790
2330 CALL SOUND(500,800,2)
2340 ST$="LEGEND 1"
2350 GOSUB 770
2360 D2$=D$
2370 ST$="LEGEND 2"
2380 GOSUB 770
2390 D3$=D$
2400 D$=D2$
2410 R=23
2420 Y=7
2430 GOSUB 1530
2440 D$=D3$
2450 R=24
2460 Y=7
2470 GOSUB 1530
2480 CALL KEY(0,K,S)
2490 IF S=0 THEN 2480
2500 IF K=80 THEN 2560
2510 IF K=78 THEN 630
2520 IF K=83 THEN 2540
2530 GOTO 2480
2540 CALL CLEAR
2550 END
2560 OPEN #1:DEV$
2570 FOR X=1 TO 24
2580 FOR P=1 TO 32
2590 CALL GCHAR(X,P,Z)
2600 IF Z<96 THEN 2620
2610 Z=35
2620 IF (Z<91)+(Z>94)THEN 26
40
2630 Z=95
2640 PRINT #1:CHR$(Z);
2650 NEXT P
2660 PRINT #1:""
2670 NEXT X
2680 CLOSE #1
2690 GOTO 1820
```

==========

TIBase Topic - TUTORing #02
by Tony Kleen, Guilford TI99er Users Grp

Article 09; Copyrighted August 1991
Reprints are encouraged! Let's use it.

==========

This article deals with creating a database, adding records to a database, and editing records that already exist on a database.

''''''

us. After entering the directive CREATE ADDRESS at the .dot prompt, the following screen is displayed:

```
arrows to move, enter to advance

FIELD  DESCRIPTOR  TYPE  WIDTH  DEC

  1    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
  _    _________   _     ___    __
```

It is expected that you enter a value, and then hit 'enter' to advance to the next value. DESCRIPTOR is nothing more than the 'field name'. TYPE and WIDTH have been described previously. DEC is the number of decimal places for a (N)umeric type of field. After you have entered a field's values, that is, after you enter the last value on a line, TIBase will advance to the next line.

What follows is my completed structure for the ADDRESS database. I have completed entry of my six fields, and have entered a <F8>, function 8, to inform TIBase of my completion. TIBase then creates an empty database on the diskette, and then prompts you as to whether or not you wish to enter data at this time.

```
FIELD  DESCRIPTOR  TYPE  WIDTH  DEC

  1    RECNUM       N     006    00
  2    ADDR_LN_1    C     024
  3    ADDR_LN_2    C     024
  4    CITY         C     018
  5    STATE        C     002
  6    ZIPCODE      C     010

do you wish to input data now? (Y/N):
?
```

If you wish to populate your database at this time, simply enter (Y)es. TIBase will then prompt you with it's APPEND interactive editor, which, for my database, looks like the following:

```
                APPEND

RECNUM      _____                    000
ADDR_LN_1   ____________________
ADDR_LN_2   ____________________
CITY        ____________
STATE       __
ZIPCODE     ________

000 1 ADDRESS 00000/00000eof
```

Looks very similar to the STRUCTURE, doesn't it? Each field of the record is presented in the order you described that field to the STRUCTURE. To the right of each field is an entry area for you data. The size of this area is dependent upon the size of the field, as you entered it on the STRUCTURE. If the size of the field is greater than 26, then, as you enter data for that field, the editor will tab to the right 10 characters to allow you to continue entering data. That's the reason for the three zeroes to the right of the RECNUM entry. As you enter data, that number is incremented to show you what character number you are entering. This number is not of much value if your field sizes are less than 26 characters, but is necessary when enter data for field sizes greater than 26. As you enter data, you can use the arrows to move left and right, and the insert and delete keys.

3.2.4 APPENDING DATA

As each record is filled with data, either an EXECUTE, <F8>, or an ENTER at the last field will cause the data to be stored and another blank record be made available for appending data.

When the APPEND session is complete, an ESCAPE, <F9>, will return to the main screen.

(*)

Now then, if you don't wish to populate the data base after you have entered the structure, you simply answer (N)o to the 'do you wish to input data now?' question; and the CREATE session will return to the main screen.

At this point, you have CREATEd a database, and may or may not have populated that database with data.

USE

Once a data-base has been created it is made active via the USE directive.

USE (filename) i.e. USE DSK2.ADDRESS

If no device is specified, the default device in DATDISK will be utilized. The USE directive will cause the named data-base to become active in the currently selected slot. If a data-base is already active in the current slot, either a CLOSE must be issued to close the active data-base or another slot must be selected.

When you have completed using the data-base, it may be closed via the CLOSE directive.

Note that all databases must be closed prior to terminating the TI-BASE program or data may be lost.

The QUIT directive will close all active databases prior to terminating TI-BASE.

----------(*)

After you have CREATEd a database, this is the only way you can USE that database, ie, issue the USE directive. Seems logical enough, right?

If you don't specify a specific disk drive number, or disk name, TIBase will default to the DATDISK that you have previously specified. That discussion on 'slots' will be covered later, in the SELECT discussion. Right now, we're doing everything in slot #1.

I probably should have covered this earlier. When you wish to end your TIBase session, you enter the QUIT directive. This directive will CLOSE any opened databases prior to exiting the TIBase application software.

You'll see an example of the USE directive after we introduce APPEND and EDIT.

\\

APPEND

To add data to an existing data-base the APPEND directive is utilized:

APPEND (BLANK) i.e. APPEND

APPEND BLANK will add a blank record to the data-base which may then have data replaced or edited into it.

APPEND will allow the user to input data interactively from the keyboard.

----------(*)

APPEND cannot be used unless you have an active database. Remember, databases are made active by the USE directive.

If you enter APPEND, TIBase will place you in the APPEND interactive editor. This is the same editor presented to you by answering (Y)es to the 'do you wish to input data now?' after you have CREATEd you data base's STRUCTURE. We covered this just above in the CREATE section.

APPEND BLANK will add JUST ONE record to the database. You do not get placed into the APPEND interactive editor. All the field values are space filled.

\\

3.2.4 APPENDING DATA

APPEND BLANK will append a blank record to the current database and return immediately. This allows blank records to be appended in the command language, and data to be replaced into the records programmatically.

----------(*)

I'll cover APPEND BLANK when we get to command files, which is considered programming.

\\

EDIT

The EDIT directive allows the user to enter data interactively.

EDIT (n) i.e. EDIT 5

The specified record will be retrieved and presented for editing. If no record number is specified, the current record will be retrieved.

After editing the record, it may be saved by an execute (F8) or by an ENTER at the last field. Paging (F5) or (F6) simply retrieves new records and does not save them.

When the desired data has been edited ESCAPE (F9) will return to the main screen. HELP (F7) is available to define the allowable function keys.

----------(*)

EDIT cannot be used unless you have an active database. Remember, databas are made active by the USE directive.

If you enter EDIT, TIBase will place you in the EDIT interactive editor. EDIT editor is very similar to the APPEND editor. With APPEND, you're always adding a new record to the end of the database. With EDIT, you're always modifying an old record, one that already exists. Since you're editing existing records, this editor allows you to 'page forward', <F5>, or to 'page backward', <F6>.

EXECUTE, <F8>, and ESCAPE, <F9>, function the same as in the APPEND editor.

Okay, now that I've introduced CREATE, USE, APPEND, and EDIT; let's give an example:

We're already done the following:

```
SET DATDISK DSK.DATA
CREATE ADDRESS
```

We've also entered the database's STRUCTURE, which can be reviewed in the above CREATE section. When I created this structure, I answered (N)o to the 'do you wish to input data now?' question. I now wish to enter data. Here's what's needed:

```
USE ADDRESS
APPEND
```

After entering the APPEND directive, I'm presented the APPEND interactive editor. At this point, I enter the following records. Remember, the APPEND editor presents me another blank record after I enter the last field on each record. To conclude entry, I simply ESCAPE, <F9>, the APPEND editor.

\\

APPEND

```
RECNUM      000000                    006
```

```
ADDR_LN_1  3239 High Cliffs Road___  021
ADDR_LN_2  ________________________  000
CITY       Pfafftown_____________   009
STATE      NC                       001
ZIPCODE    27040-9552               009


        000 1 ADDRESS 00000/00000eof
-------------------------------------------

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
             APPEND

RECNUM     000001                   006
ADDR_LN_1  279 Sara Lane_________   013
ADDR_LN_2  Apartment #B-9________   014
CITY       Winston-Salem____        013
STATE      CN                       001
ZIPCODE    27106____                005

        000 1 ADDRESS 00000/00001eof
-------------------------------------------

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
             APPEND

RECNUM     000002                   006
ADDR_LN_1  4874 Spainhour Mill Road 024
ADDR_LN_2  ________________________  000
CITY       Tobaccoville_____        012
STATE      NC                       001
ZIPCODE    27045____                005


        000 1 ADDRESS 00001/00002eof
-------------------------------------------
```

At this point, I have entered the ESCAPE, <F9>, to exit the APPEND interactive editor. This returns us to the .DOT prompt. Our database is still in use: we have not yet CLOSEd it nor have we QUIT.

If you notice above, I have incorrectly entered CN for Winston-Salem. We all know this city resides in North Carolina (NC). We need to edit our record.

At the .DOT prompt, I enter EDIT, to enter the EDIT interactive editor. This presents us with the 'current' record, the last record we accessed. In this case, the 'current' record is the last record we APPENDed:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
             EDIT

RECNUM     000002                   000
ADDR_LN_1  4874 Spainhour Mill Road
ADDR_LN_2
CITY       Tobaccoville
STATE      NC
ZIPCODE    27045


        000 1 ADDRESS 00002/00003eof
-------------------------------------------
```

We want the record previous to this one. We have two ways to get to this record. We could ESCAPE and EDIT 1; or we could simply PAGE BACKWARD, <F5>. I take the simplest method, <F5>.

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
             EDIT

RECNUM     000001                   000
ADDR_LN_1  279 Sara Lane            000
ADDR_LN_2  Apartment #B-9           000
CITY       Winston-Salem            000
STATE      NC                       001
ZIPCODE    27106


        000 1 ADDRESS 00001/00002eof
-------------------------------------------
```

What I've done here is hit the ENTER key four times to get to the STATE field entry. Then I keyed over my mistake. At this point, I would want to EXECUTE, <F8>, my change.

After I have EXECUTEd my change, the EDIT editor presents me with the next record. At this point, I have made my change, and am ready to ESCAPE, <F9>.

After I have ESCAPEd, I am once again presented the .DOT prompt. I have now CREATEd and populated my ADDRESS database. I am now ready to QUIT, quite literally.

END of DISCUSSION. We have CREATEd, USEd, APPENDed, and EDITed our ADDRESS database. What more could we possibly hope to do? LOTS!! Later.

FOOTNOTE: (*) These sections were copied from Dennis Faherty's TUTOR diskette, or the TIBase Users Guide.

LOWERCASE

by Bob Carmany

Here is a short program that will produce a series of true lowercase letters in your programs instead of the "small caps" of the normal TI character set. When you RUN the program, the character set will appear across the bottom of the screen. If you want to use the characters in one of your own programs, simply erase line 100, 180, and 190 and resequence the remaining lines so that they are at the beginning of your program. Then, whenever you select a lowercase letter, it will appear as the corresponding letter in the demo instead of a small capital letter.

```
100 CALL CLEAR :: FOR U-33 TO 94 :: CALL CHARPAT(U,U$):: CALL CHAR(U,SEG$(U$,3,1 4)):: NEXT U

110 CALL CHAR(97,"000038043C243C",98,"20203824242438",99,"00001824202418",100,"0 4041C2424241C")

120 CALL CHAR(101,"00003C243C203C",102,"081410103C101",103,"00001C24241C0418",10 4,"20203824242424")

130 CALL CHAR(105,"0010001010101",106,"000800080808083",107,"20202428302824",108 ,"3010101010101")

140 CALL CHAR(109,"00007854545454",110,"00003824242424",111,"00001824242418",112 ,"0000382424382020")

150 CALL CHAR(113,"00001C24241C0404",114,"00003C2420202",115,"00003C203C043C",11 6,"00103810101018")

160 CALL CHAR(117,"0000242424241C",118,"0000242424181",119,"00005454545428",120, "00004428102844")

170 CALL CHAR(121,"000024241408102",122,"00003C0810203C")

180 PRINT "abcdefghijklmnopqrstuvwxyz"

190 FOR DE=1 TO 2000 :: NEXT DE :: END
```